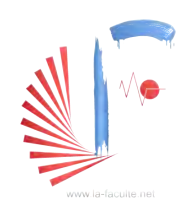









You
Tube

Informatique :

* Algoritme :

a) Organigrame : C'est résoudre un problème avec des symbole graphique

(ovale) : début / fin

(parallélogramme) : lire / écrire

(rectangle) : le calcule = → :=

non (losange : cond) oui — text

≥ ≤ ≠ a < x < b

≥ → >= ; ≤ → <= ; ≠ → < > ; x > a et x < b

* Chaque organigrame commence par un début, après début il faut lire les données mais avant de lire il faut donner des nom pour les variable (c'est un ensemble de caractaire tel que le 1 caractaire doit être une lettre), après il faut traiter le but et afficher le résultats

exo 1 : écrire un organigrame qui calcule le minimum de deux valeur

début
lire (a,b)
y = a - b
y > 0 : Non → Min := a ; oui → Min := b
écrire (Min)
fin

* déroulez ou éxécutez cette organigrame pour les valeurs 15 et 17

le déroulement : - il faut déssiner un tableau tel que le nombre de colone égal le nombre de variable utilisé

a	b	y	Min
15	17	-2	15
17	15	2	15
15	15	0	15

2ème méthode :

déroulement :

a	b	Min
15	17	15
17	15	15
15	15	15

* exo 2 : écrire un organigramme qui calcule la moyenne de 3 notes

$$Moy := \frac{x+y+z}{3}$$

dérouler cette organigramme pour les valeurs 15, 17, et 16

x	y	z	Somme	Moy
15	17	16	48	16

x	y	z	Moy
15	17	16	16

* exo 03 : écrire un organigramme qui calcule le maximum de 3 valeur

* les valeur 15, 17 et 26

x	y	z	Max
15	17	26	17
1			26

* écrire un organigrame qui calcule les solutions d'une equation de degré 2. معادلة درجة 2

début
lire (a,b,c)
a=0 — oui / Non
Non : del := (B*B) - (4*B*C)
del > 0 — oui : $x_1 = (-B - \sqrt{del}) / (2*a)$; $x_2 = (-B + \sqrt{del}) / (2*a)$; écrire (x_1, x_2)
del > 0 — Non : Del = 0 — oui : $X_3 = -b/2*a$; écrire (X_3) ; Non : écrire (Pas de sol)
a=0 oui : b=0 — Non : $X_4 = -c/b$; écrire (X_4)
b=0 oui : c=0 — oui : écrire ("R") ; Non : écrire (Pas de sol)
Fin

* dérouler cette organigrame pour les valeurs 4,5,1.

a	b	c	del	X_1	X_2	X_3	X_4
4	5	1	9	-1	-1/4		

* écrire un organigrame qui calcule le nombre de valeur positif dans une suite de "n" valeurs

début
lire (N)
i := 1
nb := 0
i <= N — oui : lire (val) ; val > 0 — oui : nb = nb + 1 ; Non : i := i + 1 (retour au test i <= N)
i <= N — Non : écrire (nb)
Fin

* dérouler cette organigrame pour N=6. les valeurs 20, -17, 15, 11, 0 et 3.

N	i	nb	val
6	1	0	20
	2	1	-17
	3	2	15
	4	3	11
	5	4	0
	6		3
	7		

* écrire un organigrame qui calcule la somme des val superieur à 5 dans une suite de N val

début → lire → i := 1 ; nb := 0 → i <= N
- Non → écrire(Nb) → Fin
- Oui → lire(val) → val > 5
  - Oui → nb = nb + val → i := i + 1
  - Non → i := i + 1

N	i	Nb	val
6	1	0	20
	2	20	-17
	3	35	15
	4	46	11
	5		0
	6		3
	7		

* écrire un organigrame qui calcule le produit des valeurs > 5 et < 20 dans une suite de n valeurs

début
lire (N)
i := 1
P := 1
i <= N
non
écrire (P)
Fin
oui
lire (val)
val > 5 et val < 20
Non
oui
P := P * val
i := i + 1

N = 5 / 10, 15, -5, 3, 22

N	i	P	val
5	2	1	10
	3	10	15
	4	150	-5
	5		3
	6		22

* ecrire un organigrame qui calcule la some des val (-) et le produit des val > 5 dans une suite de N val.

début
lire (N)
i := 1
P := 1
S := 0
i ≤ N
Non
écrire (P, S)
Fin
Oui
lire (N)
val > 5
Non
val < 0
oui
S = S + val
Non
i = i + 1
Oui
P = P * val

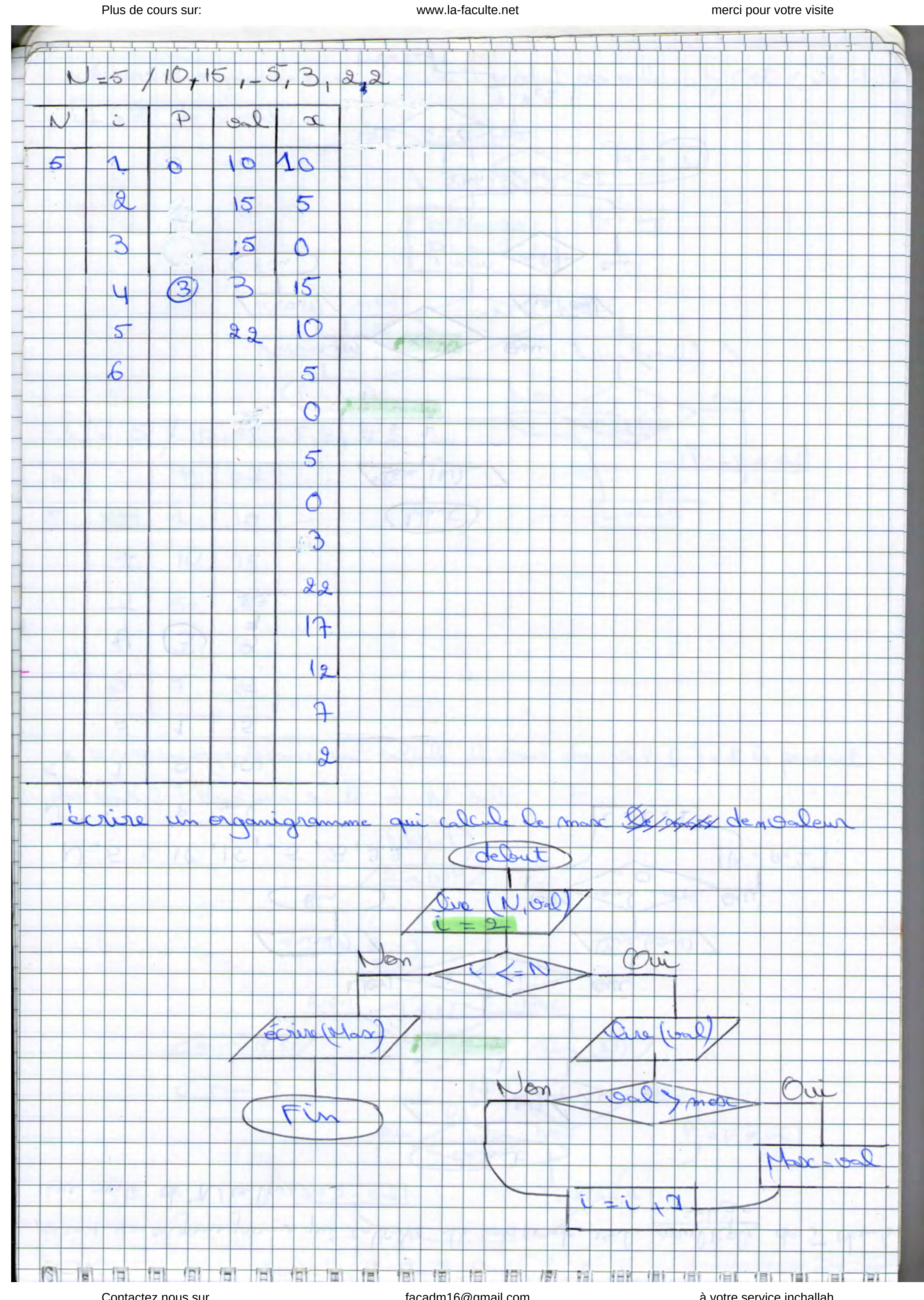
écrire un organigrame qui calcule le nbre de val multiple de 5 dans une suite de N valeur

début
Lire (N)
i := 1
P := 0
i <= N — Non: écrire(P), fin — oui: Lire (val)
val mod 5 = 0 — oui: P := P+1 — Non
i := i+1

N=5 : 10, 15, -5, 3, 22

N	i	P	val
5	1	0	10
	2	1	15
	3	2	-5
	4	(3)	3
	5		22
	6		

début
Lire (N)
i := 1
p := 0
i <= N — Non: écrire (P), Fin — oui: Lire (val)
val > 0 — non: x := -val — oui: x := val
x > 5 — oui: x := x - 5 — Non
x = 0 — oui: P := P+1 — Non
i := i+1



N=5 / 10, 15, -5, 3, 22

N	i	P	val	x
5	1	0	10	10
	2		15	5
	3		15	0
	4	(3)	3	15
	5		22	10
	6			5
				0
				5
				0
				3
				22
				17
				12
				7
				2

- écrire un organigramme qui calcule le max ~~de~~ de valeur

* Déroulement : $x = 2$ $a = 2$ $N = 8$

x	a	N	Q	F	T	i	P
2	2	8	1	6	1	1	0
			4	24	2	2	14
			16	120	4	3	14 + 136/4
			64	720	8	4	+ 784/8

Déroulez pr : $x = 2$ $a = 3$ $N = 9$

x	a	N	Q	F	T	i	P
2	3	9	1	6	1	1	0
			4	24	3	2	28/3
			16	120	9	3	136/3 + 28/3
			64	720	27	4	+ 784/27
			256	5040	81	5	+ 5296/81

* Organigramme qui calcule :

$$P = \prod_{i=0}^{N+2} \frac{2^i + ((i+2)!)^2}{x^{2i}}$$

$2^i \to Q$, $((i+2)!)^2 \to T$, $x^{2i} \to F$

$i = 0$ $Q = 1$ $T = (2!)^2$ $F = 1$

$i = 1$ $Q = 2$ $T = (4!)^2$ $F = x^2$

$Q = Q * 2$ $T = T * (i+2)^2$ $F = F * x^2$

* Les val Ini

$i = -1 \to Q = \frac{1}{2}$ $T = 1$ $F = \frac{1}{x^2}$

début

lire (x, N)

Q := 1/2
T := 1
F = 1/(x * x)
P = 1
i = 0

i ≤ N + 2 — Non / Oui

Non: ecrire (P) → Fin

Oui:
Q = Q * 2
T = T * (i + 2) * (i + 2)
F = F * x * x
P = P * (Q + T) / F
i = i + 1

déroulez cette organigrame x = 2 N = 2

x	N	Q	T	F	i	P
2	2	1/2	1	1/4	0	1
		1	4	1	1	5
		2	36	4	2	$5 \times \frac{38}{4}$
		4	576	16	3	$5 \times \frac{38}{4} \times \frac{580}{16}$
		8	144000	64	4	$\times \frac{14408}{64}$
		16	184000	256	5	$\times \frac{51846}{256}$

$$P = \sum_{i=2}^{50} \frac{a^{Q} + (i+5)!}{b^{2i+1}}$$

Q, T, Z

$i = 2$ $T = 7!$ $Z = b^5$

$i = 3$ $T = 8!$ $Z = b^7$

$T = T * (i + 5)$ $Z = Z * b^2$

* dérouler pour N = 5, les val = 25, 30, 15, 49, 2

N	val	i	max
5	25	2	25
	30	3	30
	15	4	49
	49	5	
	2	6	

$$P = \sum_{i=1}^{N=5} \frac{x^{2i} + (i+3)!}{a^i} = \frac{x^2 + 4!}{a} + \frac{x^4 + 5!}{a^2} + \frac{x^6 + 6!}{a^3}$$

(Q = x^{2i}, T = $(i+3)!$, F = a^i)

$i = 1$ $Q = x^2$ $T = 4!$ $F = a$

$i = 2$ $Q = x^4$ $T = 5!$ $F = a^2$

$Q = Q * x^2$ $T = T * (i+3)$ $F = F * a$

* Les valeurs Ini

$i = 0$ $Q = 1$ $T = 6$ $= 3!$ $= 1 \times 2 \times 3$ $F = 1$

$i = i + 1$

$\Sigma \rightarrow$ somme

$\Pi \rightarrow$ produit

debut

Lire (N, a, x)

Q = 1
F = 1
T = 6
i = 1
P = 0

$i <= N = 5$

Oui:

Q = Q * x * x
T = T * (i+3)
F = F * a
P := P + [(Q + T)/F]
i := i + 1

Non:

Lire (P)

Fin

* Déroulement : $x = 2$ $a = 2$ $N = 8$

x	a	N	Q	F	T	i	P
2	2	8	1	6	1	1	0
			4	24	2	2	14
			16	120	4	3	14 + 136/4
			64	720	8	4	+ 784/8

Déroulez pr : $x = 2$ $a = 3$ $N = 9$

x	a	N	Q	F	T	i	P
2	3	9	1	6	1	1	0
			4	24	3	2	28/3
			16	120	9	3	136/3 + 28/3
			64	720	27	4	+ 784/27
			256	5040	81	5	+ 5296/81

* Organigramme qui calcule :

$$P = \prod_{i=0}^{N+2} \frac{2^i + ((i+2)!)^2}{x^{2i}}$$

($2^i \to Q$, $((i+2)!)^2 \to T$, $x^{2i} \to F$)

$i = 0$ $Q = 1$ $T = (2!)^2$ $F = 1$

$i = 1$ $Q = 2$ $T = (4!)^2$ $F = x^2$

$Q = Q * 2$ $T = T * (i+2)^2$ $F = F * x^2$

* Les val Ini

$i = -1 \rightarrow Q = \frac{1}{2}$ $T = 1$ $F = \frac{1}{x^2}$

début

lire (x, N)

Q := 1/2
T := 1
F = 1/(x * x)
P = 1
i = 0

$i < N + 2$ — Non / Oui

Oui :
Q = Q * 2
T = T * (i + 2) * (i + 2)
F = F * x * x
P = P * (Q + T) / F
i = i + 1

Non :
ecrire (P)

Fin

déroulez cette organigrame x = 2 N = 2

x	N	Q	T	F	i	P
2	2	1/2	1	1/4	0	1
		1	4	1	1	5
		2	36	4	2	$5 \times \frac{38}{4}$
		4	576	16	3	$5 \times \frac{38}{4} \times \frac{580}{6}$
		8	144000	64	4	↙ $\times \frac{14408}{64}$
		16	184000	256	5	↙ $\times \frac{51846}{256}$

$$P = \sum_{i=2}^{50} \frac{a^2 + (i+5)!}{b^{2i+1}}$$

(Q au-dessus de a^2, T au-dessus de $(i+5)!$, Z sous b^{2i+1})

$i = 2$ T = 7! Z = b^5

$i = 3$ T = 8! Z = b^7

T = T * (i + 5) Z = Z * b^2

P les val ini : i = i - 1 = 1 T = 720 Z = b^3

début

Lire (a, b)

Q = a * a
T = 720
Z = b * b * b
P = 0
i = 2

i <= 50

Non: Écrire (P) → Fin

Oui:
T = T * (i + 5)
Z = Z * b * b
P = P + (Q + T) / Z
i = i + 1

---

début

Lire (x, b, N)

Q = x
T = 1/b
Z = 1
P = 0
i = 1

i <= N + 1

Non: Écrire (P) → Fin

Oui:
Q = Q * x
T = T * b * b
Z = Z * (i + 1)
P = P + (Q + T) / Z
i = i + 1

x	b	N	Q	T	j	i	P
2	2	2	2	1/2	1	1	0
			4	2	2	2	3
			8	8	6	3	$3 + \frac{16}{6}$
			16	32	24	4	$3 + \frac{16}{6} + \frac{48}{24}$

$$P = \sum_{i=1}^{N+1} \frac{x^{i+1} + b^{2i-1}}{(i+1)!}$$

???....

$i = 0 \quad Q = 1 \qquad \frac{Q \text{ donné}}{Q \text{ Calcule}} = \frac{x}{1} = x$

$T = 1 \qquad \frac{T \text{ donné}}{T \text{ Calcule}} = \frac{1/b}{1} = \frac{1}{b} = b^{-1}$

* Algorithme :

- La forme général d'un algorithme est :

```
Algorithme    nom ;
variable   }  "la partie déclarative"
début      }
 |         }
 |         }  "la partie d'action"
 |         }
fin        }
```

* La partie déclaration est :

Idf : type ;   identificateur

- il existe 2 types : "entier" et "réel"

s'il existe plusieurs variable de m̂ type, je déclare toute les variables en m̂ temps tel que je les séparent par ","

exp : x : entier }
      y : entier } x, y : entier

* La partie d'action :

a) les actions simples :

1 - la lecture : c'est lire (....) ;

2 - l'écriture : " écrire ( ) ;

b) l'action conditionnel :

- il existe 2 types de cond :

1/ si cond alor action 1

" " sinon action 2 ;

f si ; (fin si)

2/ si cond alor action ;
f si ;

* exercice 01 :

écrire un algorithme qui calcule la moyenne de 3 notes

Algorithme Ex 1 ;
variable
a, b, c, Moy : réel ;
début
lire (a, b, c) ;
Moy = (a + b + c) / 3
écrire Moy ;
Fin

15 , 14 , 17

a	b	c	Moy
15	14	17	15,33

* exercice 02

écrire un algorithme qui calcule le max de 2 valeurs

Algorithme EX 1 ;
variable
a, b, Max : entier ;
début
lire (a, b) ;
si a > b alors Max := a
sinon Max = b
f si ;
écrire (Max) ;
fin

15 , 14

a	b	Max
15	14	15

* exercice 03 :

écrire un algorithme qui calcule le min de 3 valeurs

Algorithme EX 1 ;
variable
a, b, Min, entier ;
début
lire (a, b, c) ;
si a < b alors Min = a
si non Min = b ;
f si
si Min > c alors Min = c ;
f si
écrire (Min)
fin.

(la boucle) * l'action pour : on l'utilise lorsqu'on a une répétition et le nbre de répétition est fixe

* la forme générale :

pour var = val In à val final
faire
Action ;
fait ;

fait de pour
- si var < val final
  - a) - var = val + 1
  - b) - 1 action après faire
- si var = val final
  - 1 action après fait ;

* exercice 04 :

écrire un algorithme qui calcule le nbr de val sup à 5 dans une suite de N valeurs.

Algorithme Ex 1 ;
variable
i, N, nb , val : entier ;
début
lire (N) ; nb = 0
pour i := 1 à N
faire
lire (val) ;
si val > 5 alors nb = nb + 1 ;
fsi ;
fait ;
ecrire (nb) ;
fin

N = 5 13, 8 , -5 , 0, 17

i	N	nb	val
1	5	0	13
2		1	8
3		2	-5
4		3	0
5			17

Exo 02 : Ecrire un algorithme qui calcule la somme des val dans une suite de n valeur de type entier

Algorithme , Ex 2
variable
N, i, P, val : entier
debut
lire (N) ; P := 0 ;
pour i := 1 à N
faire
lire (val) ;
si val mod 5 = 0 alors p := p + val
fait
écrire (P)
fin

i,	N,	P,	val
1	5		13

– déroulez pour N = 5 les val. 13, 20, 11, 8, -5

i	N	P	val
1	5	0	13
2		20	20
3		15	11
4			8
5			-5

* écrire un algorithme qui calcule le max de n valeur de type entier

Algorithme, Exo 2
variable
N, i, p, val : entier ;
début
lire (N, val) ; p = val ;
pour i := 2 à N
faire
lire (val)
si val > p alors p := val
fait ;
écrire (P) ;
fin

– déroulez pour N = 5. les val : 13, 20, 11, 8, -5

i	N	P	val
2	5	13	13
3		20	20
4			11
5			8
			-5

* $P = \sum_{i=1}^{N+1} \frac{x^{2i} + (i+3)!}{a^i}$ (Q = x^{2i}, T = $(i+3)!$, F = a^i)

$i = 1$ $Q = x^2$ $T = 4!$ $F = a$

$i = 2$ $Q = x^4$ $T = 5!$ $F = a^2$

$Q = Q * x^2$ $T = T * (i+3)$ $F = F * a$

- Les valeurs Ini

$i = 0$ $Q = 1$ $F = 1$ $T = 3! = 6$

Algorithme Ex 2 ;

variable

N, x, a, Q, F, T, P, i : réel

début

lire(N, x, a)

Q := 1 , F := 1 T := 6 P := 0 ;

pour i = 1 à N+1 (i := 1 tant que i <= N + 1)

faire Q := Q * x * x

T := T * (i+3)

F := F * a

P := P + (Q + T) / F

Fait; (i := i + 1)

écrire (P)

fin

* on utilise la Boucle tant que s'il existe une répitition et le nbre de répitition est fixé ou non fixé

* la forme générale de la boucle tant Que est :

tant Que

faire

Action ;

fait ;

fait de tant Que :

- si cond = V, 1 action après faire
- si cond = f, 1 action après fait ;

* écrire un algorithme qui calcule le produit des val non multiple dans une suite de N valeur de type entier.

algorithme, Ex 2 ;
variable
N, i, val, P, i : entier ;
début
lire (N) ; p := 1 ;
i := 1 ;
tantque i <= N
faire
lire (val) ;
si val mod 5 <> 0 alors p := p * val
i := i + 1
fait
écrire (P) ;
fin

N = 5 val = 13, 20, 11, 8, -5

i	N	P	val
1	5	1	13
2		13	20
3		143	11
4		143 x 8	8
5			-5
6			

Algorithme, Ex 2

variable

N, i, val, p, x : entier

début

```
lire (N) ; p=1
Pour i := 1 à N
faire
lire (val);
si val>0 alors x:=val
        si non x:=-val,
fsi
tantque x>5
faire
  x:=x-5;
  fait;
si x<>0 alors p:=p*val;
fait;
écrire(P);
```

fin

i	N	p	val	x
1	5	1	13	13
2		13	20	8
3		143	11	3
4		143x8	8	20
5			-5	15
				10
				5
				0
				11
				6
				1
				8
				3
				5
				0

Chapitre III : vecteur

- C'est un ensemble de case tel que le nbre de case est fixé

V

-15	13	12	10	0

V [3] = 12   V [5] = 0

Nom : tableau [taille] type ;

A, B : tableau [20] entier

* pour i := 1 à N
faire
    écrire (V[i])
fait ;

* i = 1 ;
tanque i <= N
faire
    lire (V[i])
    i := i + 1
fait ;

exo 1 : écrire un algorithme qui calcule le nbre de val (+) dans un tableau A de taille N tel que N <= 50 de type entier

Algorithme Exo 1 ;
variable
N, nb, i = entier
A : tableau de taille [50] entier ;
début
    lire (N) ;
    nb = 0
    pour i = 1 à N
    faire
        lire (A[i]) ;
        si A[i] > 0 alors nb := nb + 1
    fait
fin   écrire (nb) ;

* déroulez cette algorithme pour N=5 et le tableau A

-15	13	12	10	0

N	nb	i	A[i
5	0	1	-15
	1	2	13
	2	3	12
	3	4	10
		5	0

* refait l'exo avec la boucle tanque.

```
Algorithme Exo 1 ;
variable
N, nb, i : entier
A : tableau de taille [50] entier
début
    lire (N) ;
    nb =0
    i = 1
    tant que i <= N
    faire
        lire (V[i])
        si A(i) >0 alors . = nb +1
    fait ; i = i + 1
    écrire (nb);
fin
```

déroulement :

N	nb	i	A[i]
5	0	1	-15
	1	2	13
	2	3	12
	3	4	10
		5	0
		6	

Exo 2 : écrire un algorithme qui calcule la Σ des val multiple de val dans un tableau A de taille N tel que N≤20

Algorithme Exo 1;
variable
val, N, nb, i : entier
A : le tableau [20] entier.
debut
Lire (N, val);
nb = 0 ;
pour i := 1 à N
faire
Lire (A[i])
si A[i] mode val = 0 alors nb := nb + A[i]
fait ;
fin

-15	13	12	10	0

déroulement :

val	N	nb	i	A[i]
5	5	0	1	-15
		-15	2	13
		-5	3	12
		-5	4	10
			5	0

```
Algorithme Exo 1 ;
variable
x , val , N, nb , i : entier
A : tableau [20] entier ;
début
      lire (N, val);
      nb:= 0 ;
      pour i := 1 à N
      faire
       lire (A[i])
       si A[i] >0   alors x: = A [i]
                    si non x: = - A[i]
       f si ;
       tant que x >= val
       faire
        x = x - val
       fait ;
       si x := 0  alors nb: = nb + A [i];
       fait
      écrire(nb) ;
fin
```

Exo 2 : écrire un algorithme qui calcule la Σ des val

N	val	i	nb	x	A[i]
5	5	1	0	15	-15
		2	-15	10	13
		3	-5	5	12
		4	-5	0	10
		5		13	0
				8	
				3	
				12	
				7	
				2	
				10	
				5	
				0	
				0	X

* écrire un algorithme, qui calcul le max dans un tableau de taille N tel que N ≤ 50

```
Algorithme Exo 1;
variable
   i, N, max : entier
   A : tableau de taille [50] entier;
debut
      lire (N, A[1]);
      max = A[1]
      pour i = 2 à N
      faire   lire (A[i]);
              si A[i] > max alors max := A[i];
      fait;
      écrire (max);
```

dérouler : N = 6

45	13	17	11	15	50

N	i	max	A[i]
6	2	45	45
	3	50	13
	4		17
	5		11
	6		50

exo 4 : EMD 2010.

A et B de tableau de taille N tel que N<20 de type entier. écrire un algorithme qui calcule :

1 - le produit des val > 14 et < val₁ dans le tableau A

2 - remplir le tableau K tel que K(i) = A(i) + B(i) - 6

3 - remplacer les vals < val₂ par le produit dans le tableau B

(remplacer | remplir | construire) écrire avant fait

```
Algorithme Exo1;
variable
    i, N, P, val1, val2 : entier;
    A, B, K : tableau [20] entier;
début
  1ère  lire (N, val1)
        p := 1;
        pour i := 1 à N
        faire
              lire (A[i]);
              si (A[i] > 14) et (A[i] < val1) alors p := p * [A[i]];
        fait;
        écrire (p);
```

i := 1
tanque i <= N

i = i + 1

* A et B 2 tableau de taille N tel que N ≤ 20. écrire un algorithme sans utilisé la f° mode et divise.
- le produit des val multiple de val₁ ds le tableau A
- remplir le tableau B. tq le tableau B contient les val > val₂ dans le tableau B.
- remplir le tableau K t.q K(i) = A(i) div B.

{ss mode → boucle tq
mode : الباقي
dive : قسمة}

Algorithme Exo 1,
variable x, i, N, val₁, p, val₂, nb ; entier ;
A, B, V, K tableau [20] entier;
debut
lire (N, val₁); p = 1 ;
pour i = 1 à N
faire lire (A[i]) ;
si A[i] >= 0 alors x = A[i]
sinon x = - A[i] ;
f si ;
tant que x >= val₁
faire
x = x - val₁ ;
fait ;
si x = 0 alors P = P * A[i];
fait;
écrire (P) ;

2) val₂ = 5

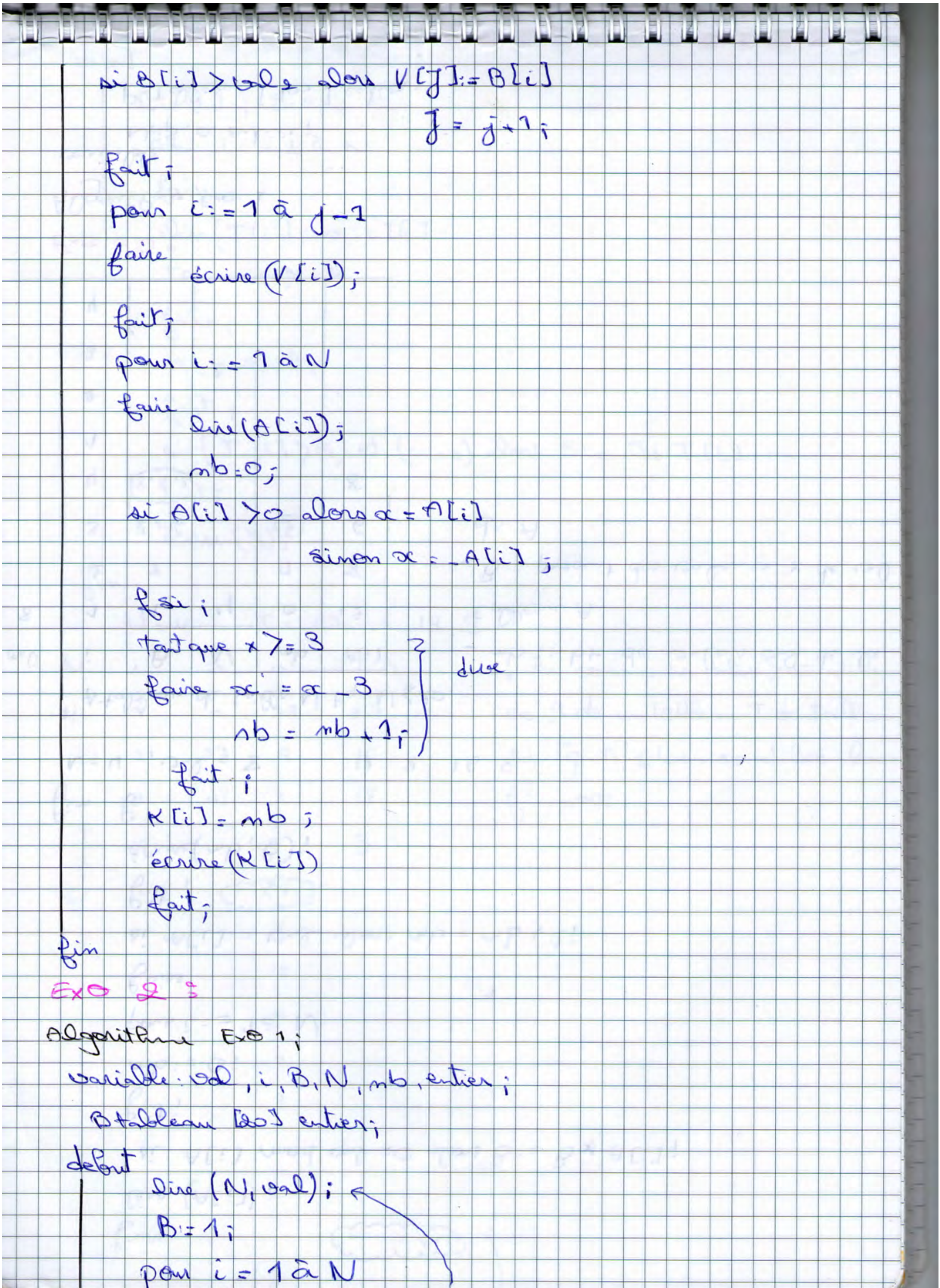
B | -1 | 13 | 12 | 3 | 8 |

V | 13 | 12 | 8 | | |

j=1 j=2 j=3 j=4

en +

j = 1;
lire (val₂) ;
pour i = 1 a N
faire
lire (B[i]) ;



```
si B[i] > val alors V[j] := B[i]
                    j = j+1;
fait;
pour i := 1 à j-1
faire
      écrire (V[i]);
fait;
pour i := 1 à N
faire
      lire (A[i]);
      nb := 0;
  si A[i] > 0 alors x = A[i]
              sinon x = -A[i];
  fsi;
  tant que x >= 3
  faire  x = x - 3
         nb = nb + 1;      } dire
   fait;
  K[i] = nb;
  écrire (K[i])
  fait;
fin
```

Exo 2 :

```
Algorithme Exo 1;
 variable : val, i, B, N, nb, entier;
  B tableau [20] entier;
debut
   lire (N, val);
   B := 1;
   pour i = 1 à N
```

faire (i < N)

lire (A[i]);

si A[i] mod val = 0 alors B := B * A[i];

fait;

nb := 0;

pour i := 1 à N

faire

si A[i] = val alors nb = nb + 1;

fait; (i < N)

écrire(nb, B);

fin f

N = 4 , val = 8 12 8 16 8

→ A tableau de taille N tq N ≤ 20.

val	i	B	N	nb	A[i]
8	1	1	4	0	12
	2	8		1	8
	3	8 x 16		(2)	16
	4	(8x16x8)			8
	1				
	2				
	3				
	4				

_ nb : nbr de valeur val ds le tableau A

_ B : produit des valeurs x de val ds A

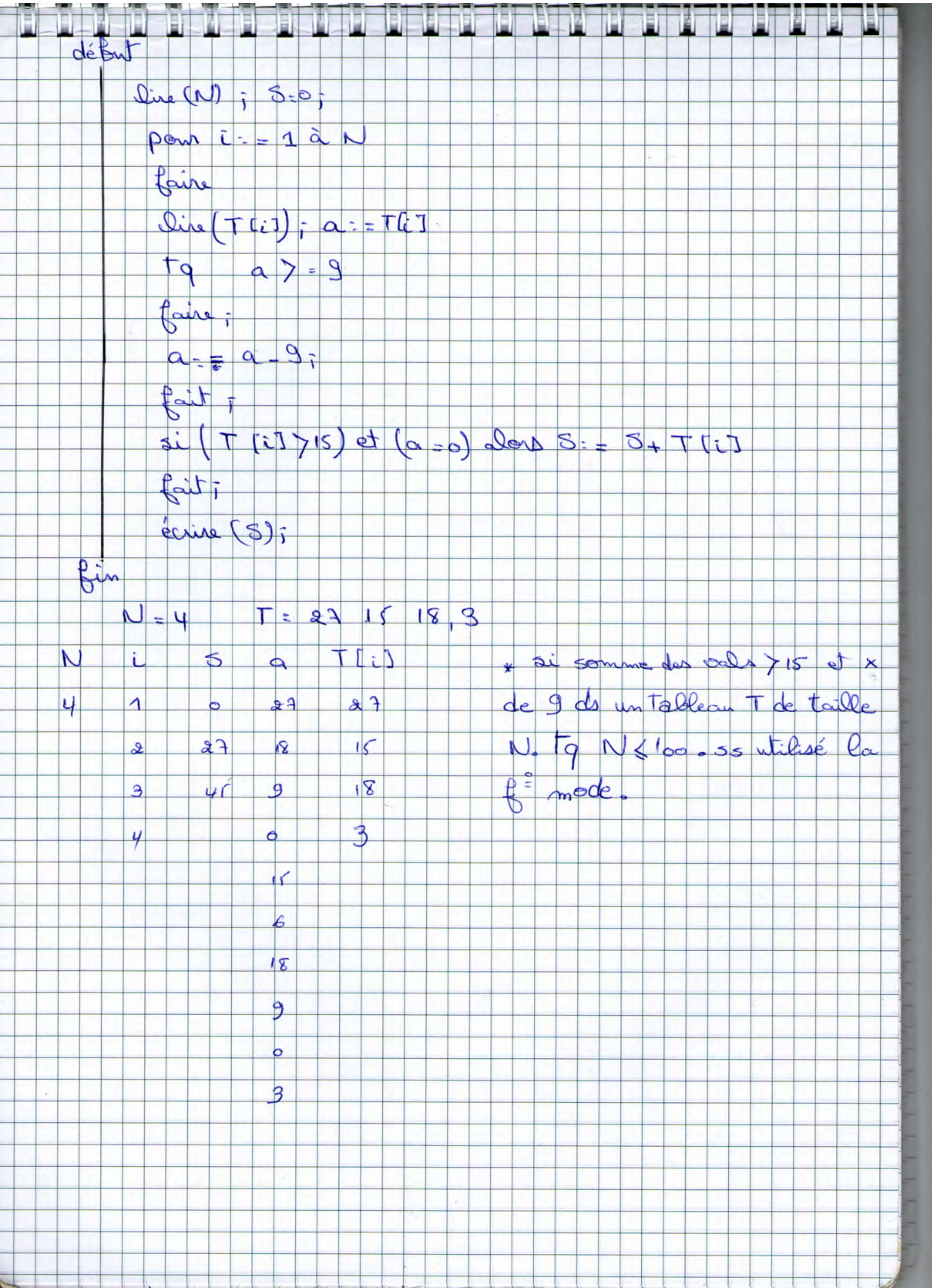
Exo 03 :

Algorithme Exo 1

variable

N, i, S, a entier;

P : tableau [100] entier;



```
début
  lire (N) ; S:=0 ;
  pour i := 1 à N
  faire
  lire (T[i]) ; a := T[i]
  Tq   a >= 9
  faire ;
   a := a - 9 ;
  fait ;
  si (T[i] > 15) et (a = 0) alors S := S + T[i]
  fait ;
  écrire (S) ;
fin
```

N = 4   T = 27 15 18, 3

N	i	S	a	T[i]
4	1	0	27	27
	2	27	18	15
	3	45	9	18
	4		0	3
			15	
			6	
			18	
			9	
			0	
			3	

* si somme des vals > 15 et × de 9 ds un Tableau T de taille N. Tq N ≤ 100. ss utilisé la f° mode.

* Les matrices :

Algorithme Exo 1 ;

$$A = \begin{bmatrix} 3 & 15 & 8 \\ 11 & 20 & -3 \end{bmatrix}$$

A [1,2] = 15

A [2 ,3] = -3

Nom: tableau [ligne ,col] type ;

A, B : tableau

ligne par lig

```
pour i := 1 à N ;
   faire(i) pour j := 1 à M ;
      faire(j)
         lire (A[i,j]);
      fait ;(j)
   fait ; (i)
```

col par col

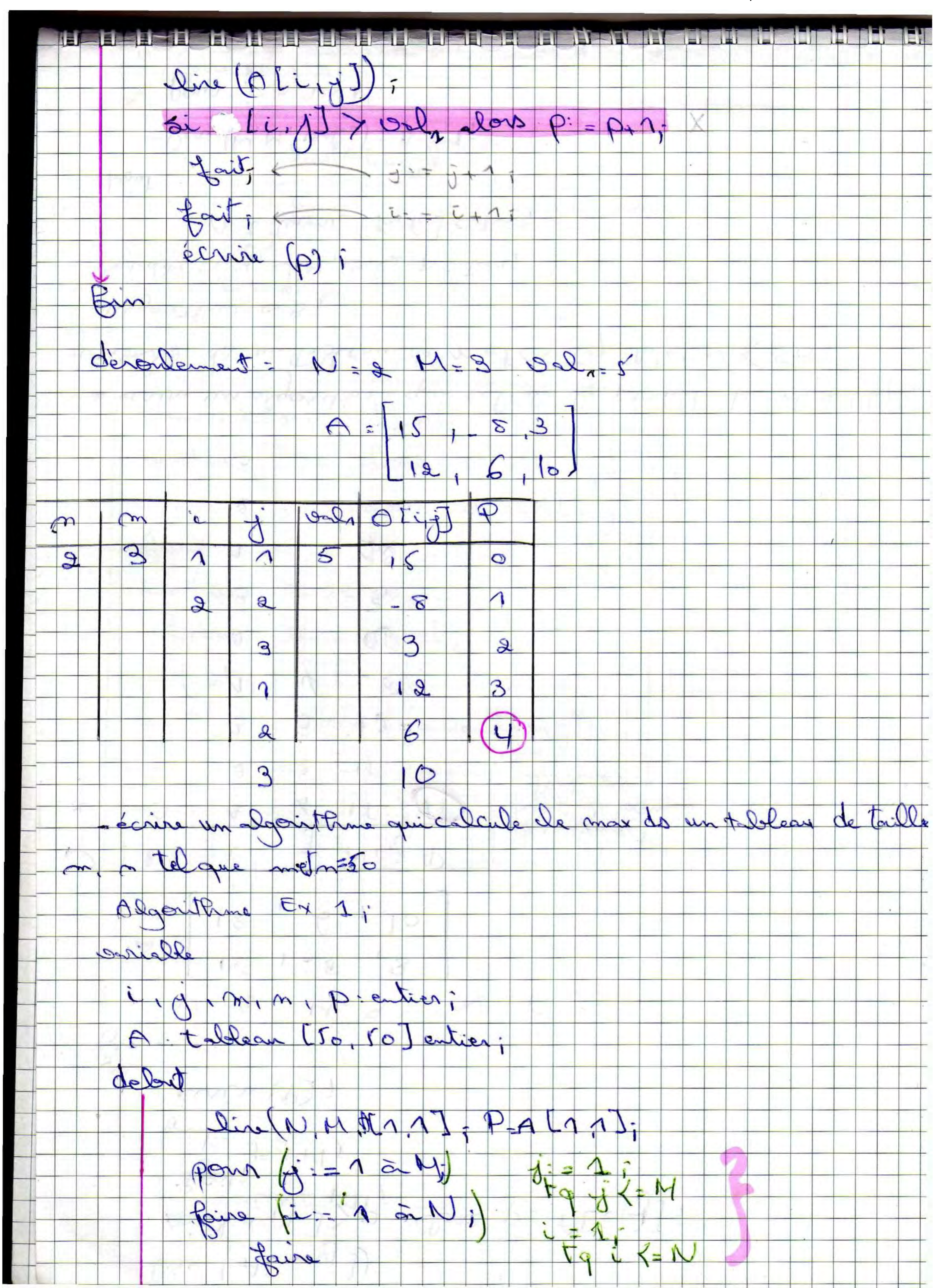
```
pour j := 1 à M ;
faire(j) pour i := 1 à N ;
      faire(i)
         lire(A[i ,j]);
   fait ;(i)
   fait ;(j)
```

exercice 1 :

- ecrire un algorithme qui calcule le nbr de val > à val1 ds un tableau de taille n et m Tq : n et m < 50

```
Algorithme Exo 1 ;
 variable
    i, j,N,M val , P : entier ;
    A:tableau [50, 50] entier,
 debut
    lire (N, M, val1) ; P = 0 ;
    pour i : = i à N ;          i = 1   tantque i <= N
    faire pour j = 1 a M ;      j = 1   tant que j <= M
      faire
```

avec la boucle tq que



lire (A[i, j]) ;
si A[i, j] > val₁ alors p: = p+1;
fait; ← j: = j+1 ;
fait; ← i: = i+1 ;
écrire (p) ;
Fin

déroulement : N = 2 M = 3 val₁ = 5

$$A = \begin{bmatrix} 15, & -8, & 3 \\ 12, & 6, & 10 \end{bmatrix}$$

n	m	i	j	val₁	A[i,j]	P
2	3	1	1	5	15	0
		2	2		-8	1
			3		3	2
			1		12	3
			2		6	4
			3		10	

- écrire un algorithme qui calcule le max ds un tableau de taille n, m tel que m et n = 50

Algorithme Ex 1 ;
variable
i, j, n, m, p : entier;
A : tableau [50, 50] entier;
debut
lire (N, M, A[1,1]) ; P = A[1,1];
pour (j: = 1 à M;)
faire (i: = 1 à N;)
faire

j: = 1 ;
Tq j <= M
i = 1 ;
Tq i <= N

lire (A[i,j])
si A[i,j] > p alors p = A[i,j] ;
i := i+1 p;
fait;
j = j+1 ;
fait;
ecrire (P) ;
fin.

N = 2 M = 3

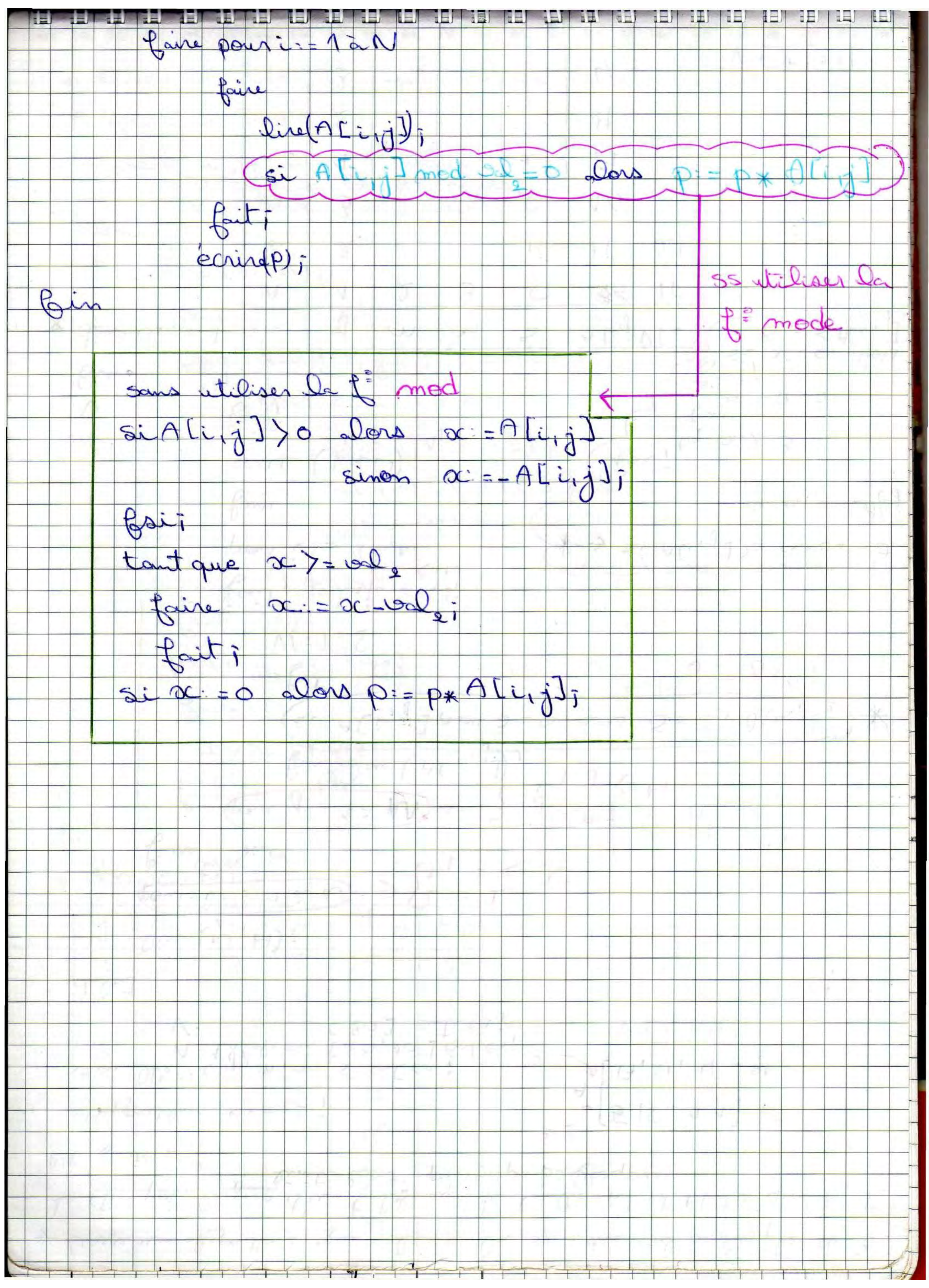
$$A \begin{bmatrix} 15, & -8, & 3 \\ 12, & 6, & 10 \end{bmatrix}$$

N	M	i	j	A[i,j]	P
2	3	1	1	15	15
		2	2	15	
		3	3	12	
		1	4	-8	
		2		06	
		3		3	
		1		10	
		2			
		3			

deroulement avec la boucle tant que.

* ecrire un algorithme qui calcule le produit des val multiple de val 2 ds un Tableau de n et m tq N et M ≤ 50

Algorithme ex 3 ;
variable : i, j, N, M, val_2, p, entier;
A : tableau [50, 50] entier;
debut
Lire (N, M, val_2) ; p := 1 ;
pour j := 1 à M


faire pour i := 1 à N
faire
lire(A[i,j]);
si A[i,j] mod val2 = 0 alors p := p * A[i,j]
fait;
écrire(p);
fin
ss utiliser la f° mode
sans utiliser la f° mod
si A[i,j] > 0 alors x := A[i,j]
sinon x := -A[i,j];
fsi;
tant que x >= val2
faire x := x - val2;
fait;
si x = 0 alors p := p * A[i,j];

* écrire un algorithme qui calcule la som des val multiple de 3 ds chaque lignes d'un tableau de taille N et M tel que N et M < 20.

(l'initialisat° après le premier faire)

(nbr de resultat = nbr de ligne)

Algorithme exo 1 ;
variable : i, j, n, m, s : entier ;
A : tableau [20, 20] entier ;
V : " [20] entier ;

S:=0 → A = [5, 15, 11, 11] 15
S:=0 → [17, 21, 4] 21

V | 15 | 21 | (sauvegarde le résultat)

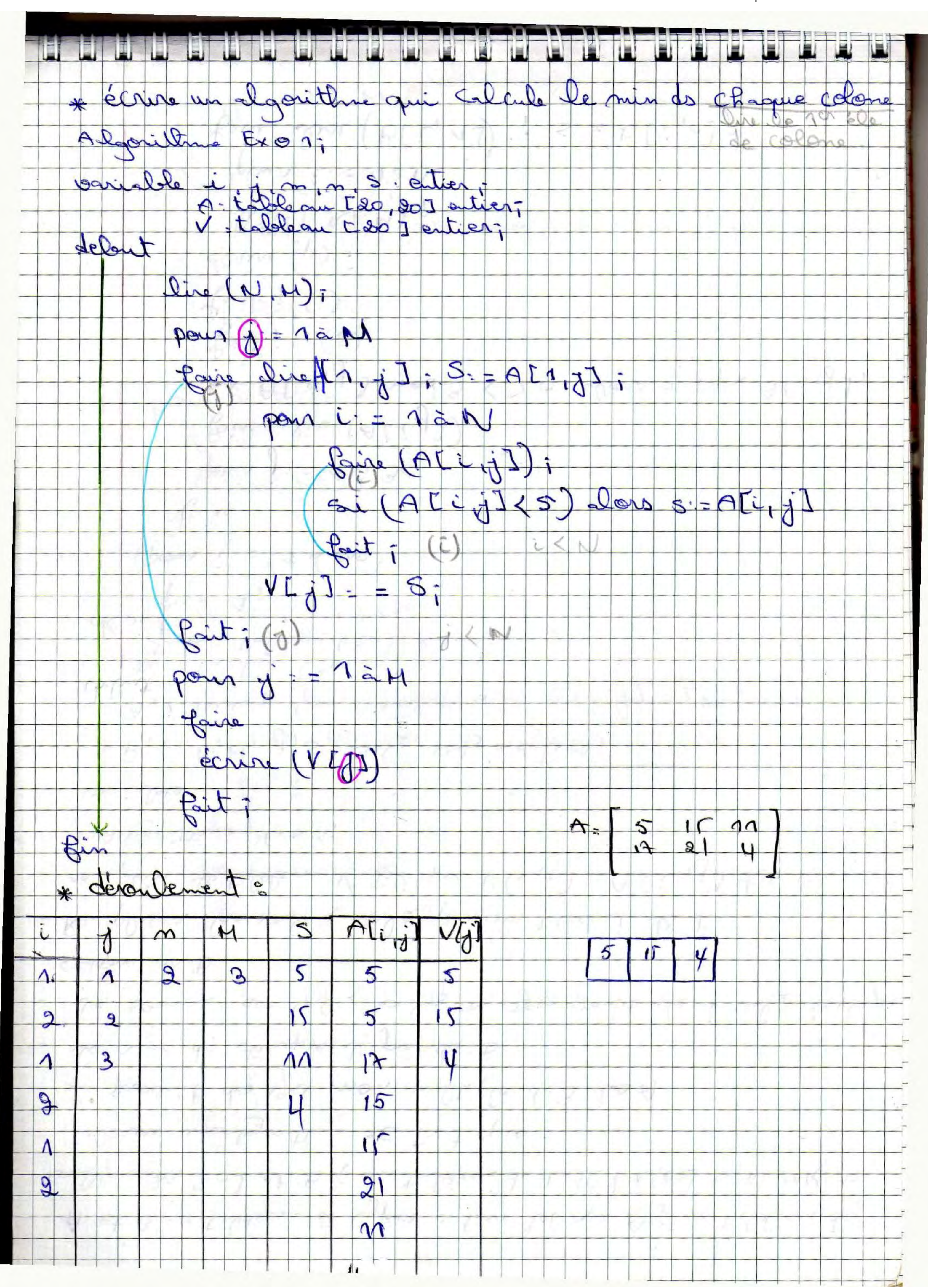
debut
Lire (N, M) ;
pour i := 1 à N → { i := 1 ; tq i <= N
faire (i) S:=0
pour j := 1 à M → { j := 1 ; tq j <= M
(j) faire Lire (A[i, j]) ;
si A[i, j] mod 3 = 0 alors S := S + A[i, j] ; *
fait ; (j) (compar le j et M) j := j + 1 ;
V[i] = S
fait ; (i) → { i = 1 ; i <= N (compar i avec N)
pour i := 1 à N
faire
ecrire (V[i]) ;
i := i + 1 ;
fait ;
Fin.

* → si A[i, j] > 0 alors x := A
sinon x := -A[i, j]
fai ;
tq x >= 3
faire
x := x - 3
fait ;
si x := 0 alors
S = S + A[i, j]

* déroulement

i	j	n	m	S	A[i, j]	V[i]
1	1	2	3	0	5	15
2	2			15	15	21
	3			0	11	
	1			21	17	
	2				21	
	3				4	

V | 15 | 21 |



* écrire un algorithme qui calcule le min de chaque colone

lire le 1er élé de colone

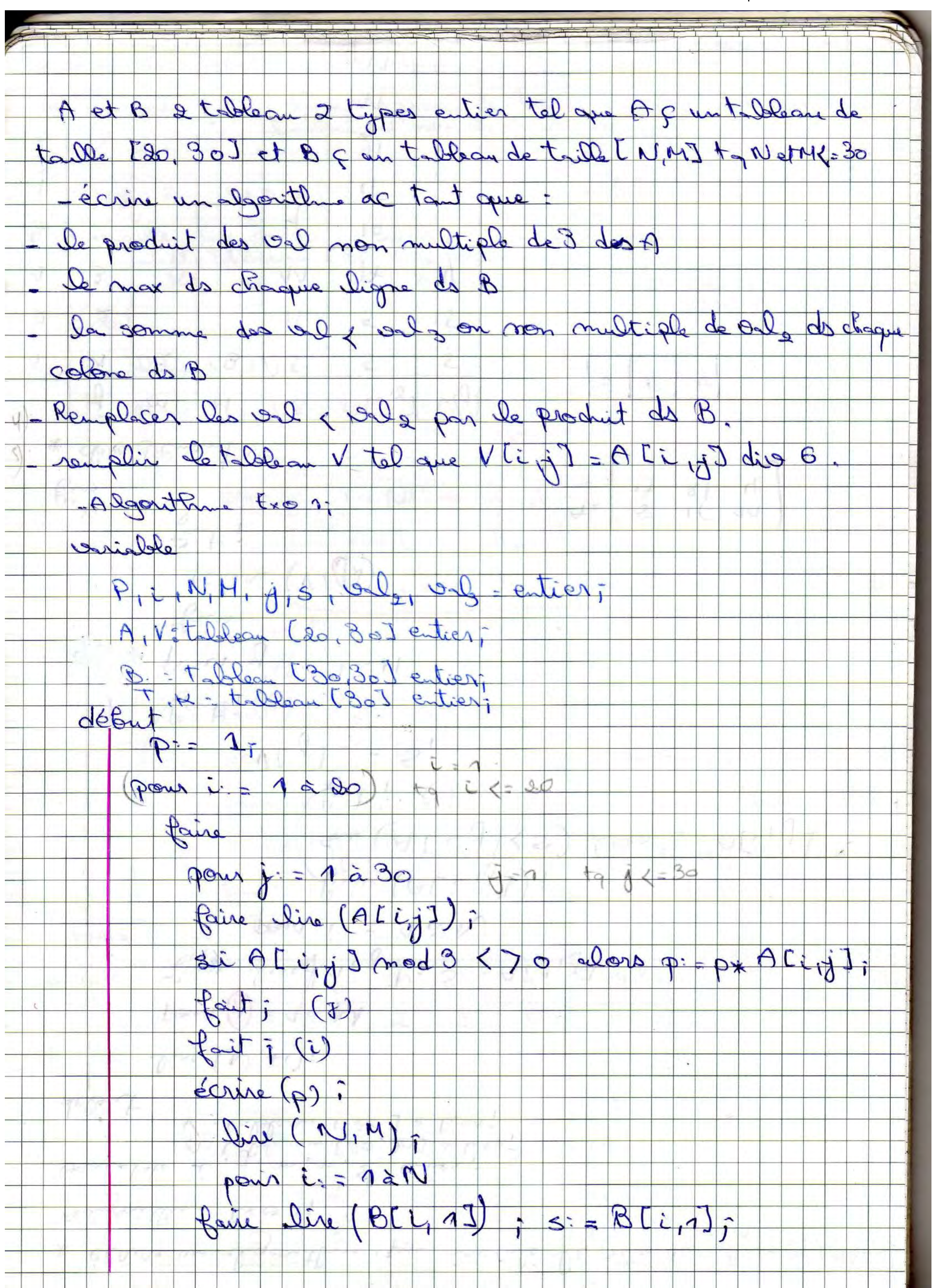
```
Algorithme Exo 1;
variable i, j, n, m, s : entier;
         A : tableau [20,20] entier;
         V : tableau [20] entier;
debut
    lire (N, M);
    pour j = 1 à M
    faire lire A[1, j]; S := A[1, j];
        pour i := 1 à N
            faire (A[i, j]);
            si (A[i, j] < S) alors S := A[i, j]
            fait; (i)      i < N
        V[j] := S;
    fait; (j)      j < N
    pour j := 1 à M
    faire
        écrire (V[j])
    fait;
fin
```

A = [5 15 11 ; 17 21 4]

* déroulement :

i	j	n	M	S	A[i,j]	V[j]
1	1	2	3	5	5	5
2	2			15	5	15
1	3			11	17	4
2				4	15	
1					15	
2					21	
					11	

5	15	4



A et B 2 tableau 2 types entier tel que A ç un tableau de taille [20, 30] et B ç un tableau de taille [N,M] tq N et M <= 30

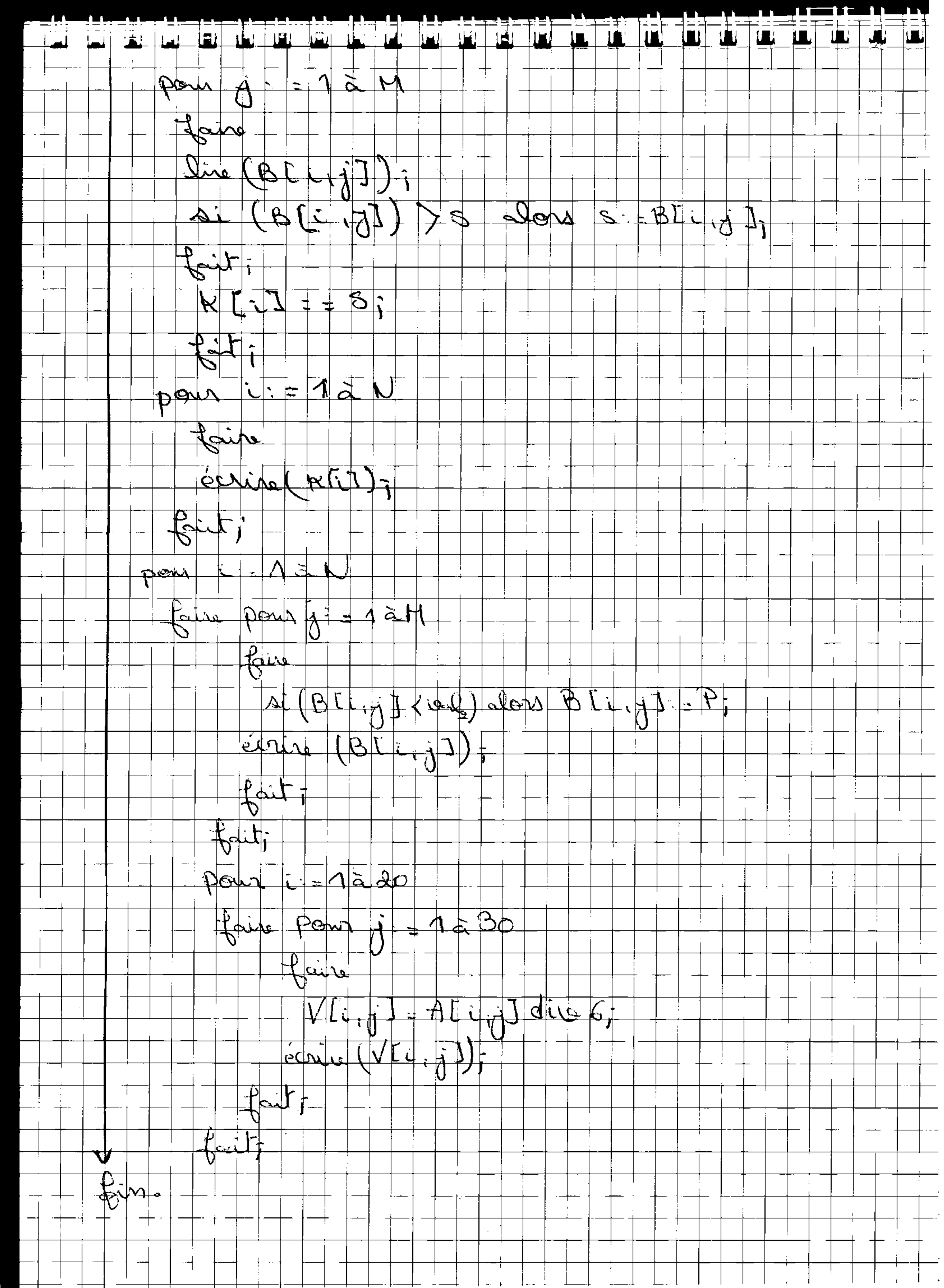
- écrire un algorithme ac tant que :
- le produit des val non multiple de 3 des A
- le max ds chaque ligne ds B
- la somme des val < val₁ ou non multiple de val₂ ds chaque colone ds B
4) - Remplacer les val < val₂ par le produit ds B.
5) - remplir le tableau V tel que V[i,j] = A[i,j] div 6.

- Algorithme Exo 1 ;

variable

P, i, N, M, j, s, val₂, val₁ = entier;
A, V : tableau [20, 30] entier;
B. : Tableau [30,30] entier;
T, K : tableau [30] entier;

début
p:= 1;
(pour i:= 1 à 20) i:=1 tq i <= 20
faire
pour j:= 1 à 30 j:=1 tq j <= 30
faire lire (A[i,j]);
si A[i,j] mod 3 <> 0 alors p:= p* A[i,j];
fait; (j)
fait; (i)
écrire (p);
lire (N, M);
pour i:= 1 à N
faire lire (B[i,1]) ; s:= B[i,1];



```
pour j := 1 à M
  faire
  lire (B[i,j]);
  si (B[i,j]) > S alors S := B[i,j];
  fait;
   K[i] := S;
  fait;
pour i := 1 à N
  faire
   écrire(K[i]);
  fait;
pour i := 1 à N
  faire pour j := 1 à M
        faire
          si (B[i,j] < val) alors B[i,j] := P;
        écrire (B[i,j]);
        fait;
     fait;
   pour i := 1 à 20
     faire pour j := 1 à 30
           faire
             V[i,j] := A[i,j] div 6;
           écrire (V[i,j]);
       fait;
     fait;
fin.
```

* on a A un tableau de taille N et M (matrice) tel que N et M ≤ 20. écrire un algorithme sans utilisé la f° mode et dire

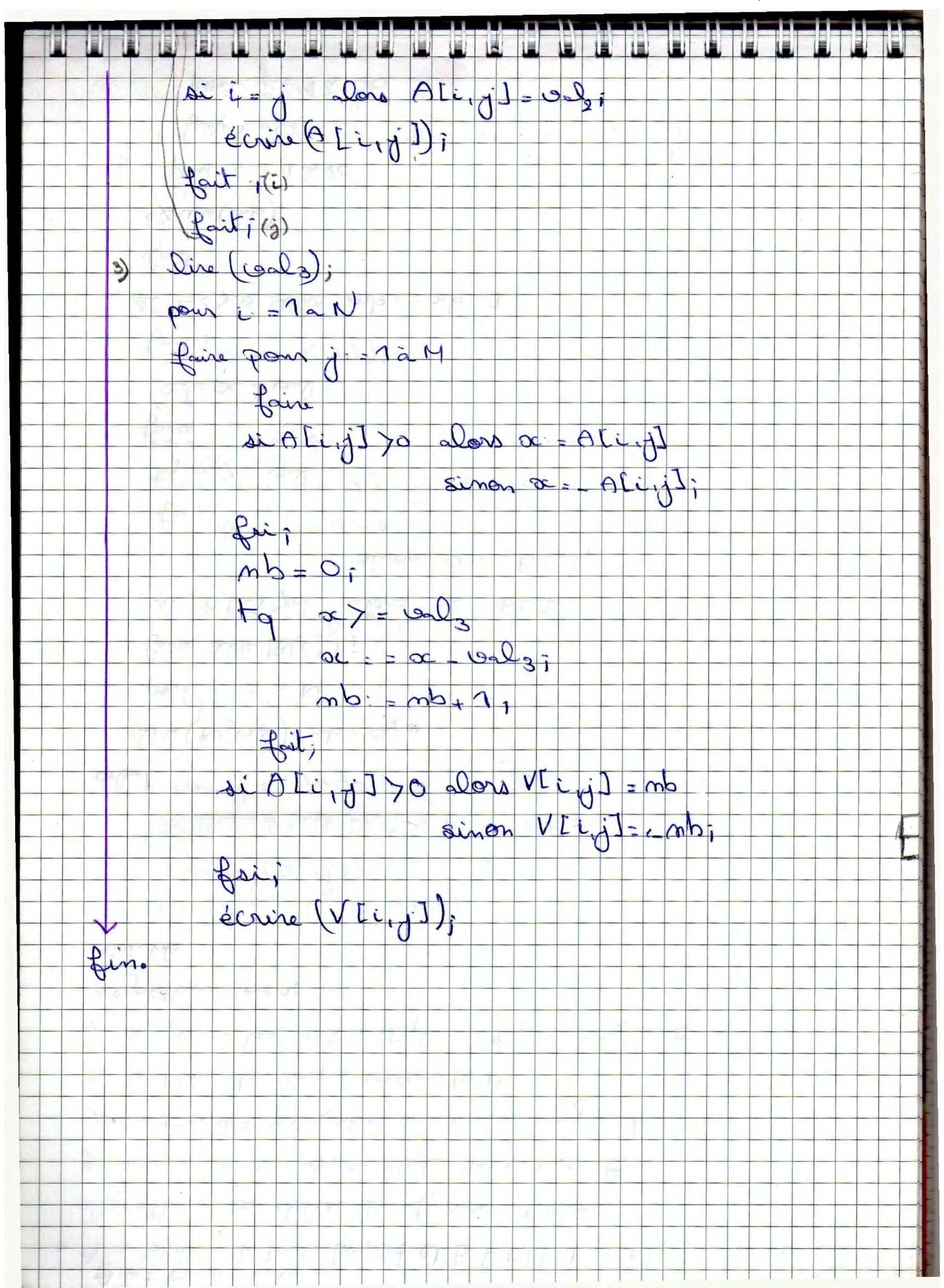
1)- qui calcule le nbr de val + ds chaque ligne

2)- remplacer les élé de la diagonale par val_2

3- remplire le tableau V tq: V[i,j] = A[i,j] dive [val_3]

dive (combien de fois existe le val_3)

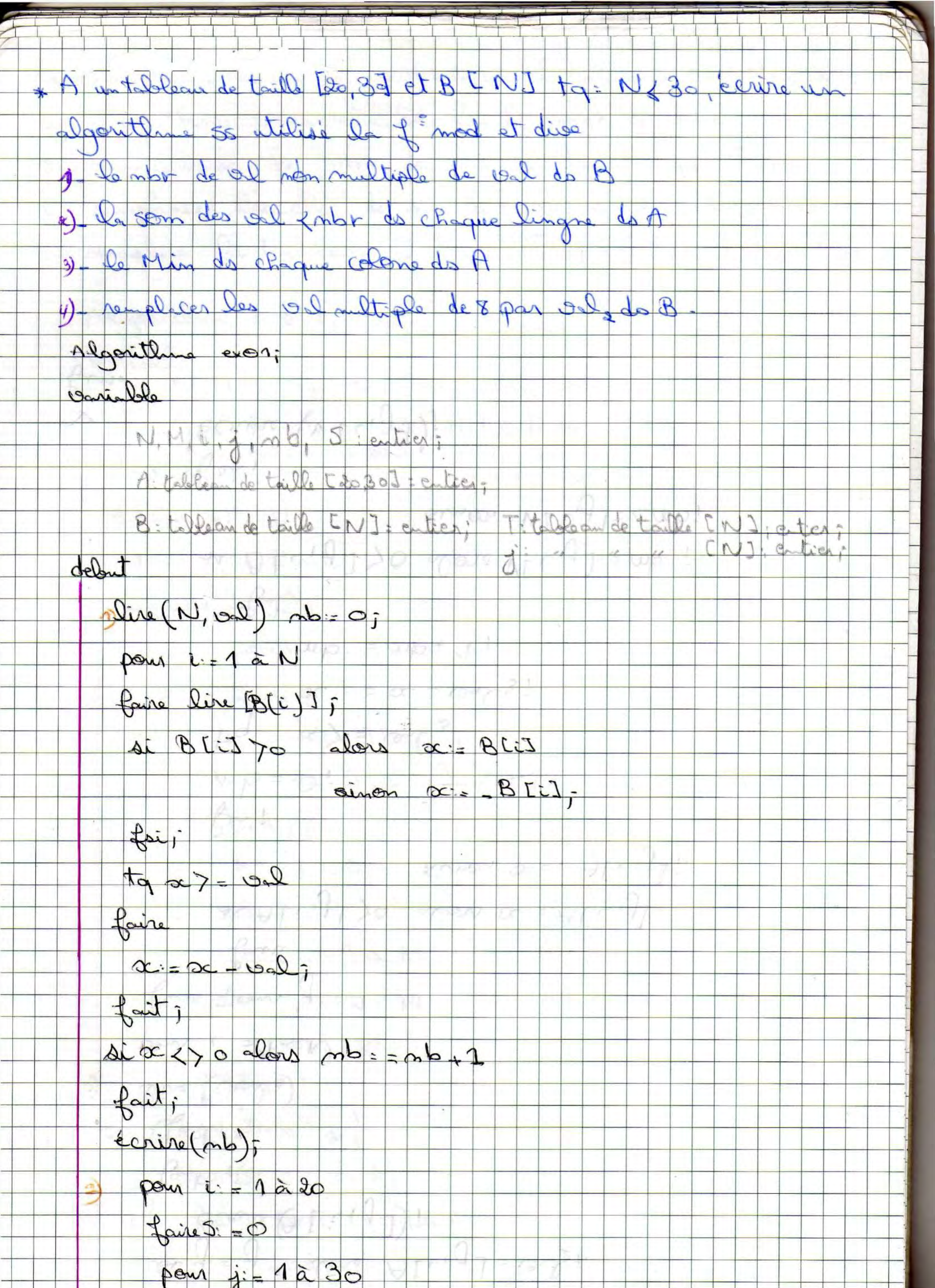
```
Algorithme Exo 1;
variable:
    N,M,i,S,j,nb,val2,val3: entier;
    A,V: tableau [20,20] entier;
    K: tableau [20] entier;
debut
1) lire(N,M);
   pour i:=1 à N
   faire(i) S:=0;
        pour j:=1 à M
        faire(j)
            lire(A[i,j]);
            si A[i,j] >0;
        fait;(j)
        K[i]:=S;
   fait(i)
   pour i=1 à N
   faire
        écrire(K[i]);
   fait;
2) lire(val2);
   pour j=1 à M
   faire(j) pour i:=1 à N
        faire(i)
```



si i = j alors A[i,j] = val₂;
écrire (A[i,j]);
fait ;(i)
fait ;(j)

3) lire (val₃);
pour i = 1 à N
faire pour j := 1 à M
faire
si A[i,j] > 0 alors x = A[i,j]
sinon x := - A[i,j];
fsi ;
nb = 0 ;
tq x >= val₃
x := x - val₃;
nb := nb + 1,
fait;
si A[i,j] > 0 alors V[i,j] = nb
sinon V[i,j] := -nb;
fsi;
écrire (V[i,j]);

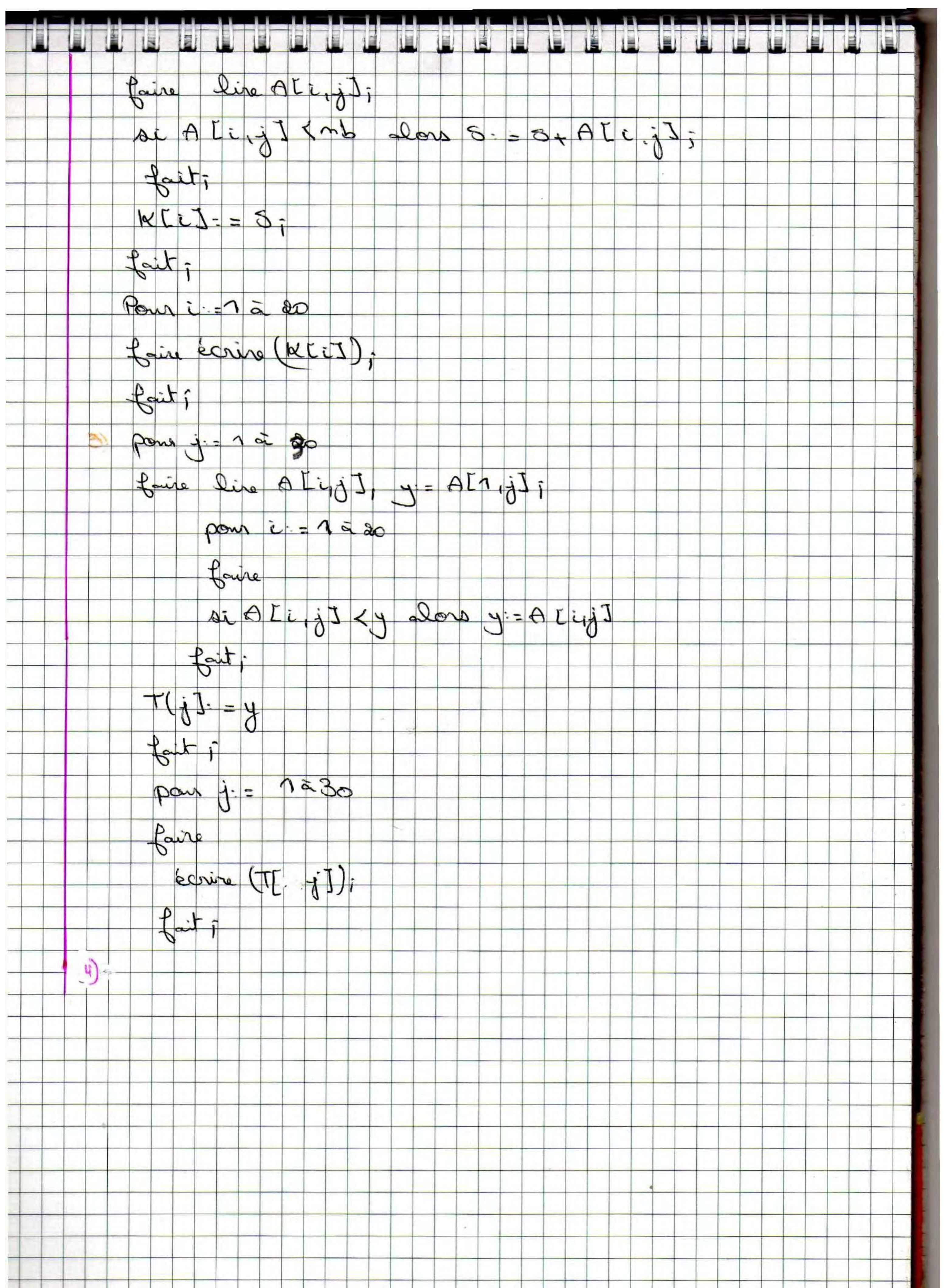
fin.



* A un tableau de taille [20,30] et B [N] tq: N ≤ 30, écrire un algorithme ss utilisé la f° mod et dire

1)- le nbr de val non multiple de val ds B
2)- la som des val ≤ nbr ds chaque lingne ds A
3)- le Min ds chaque colone ds A
4)- remplacer les val multiple de 8 par val_2 ds B.

```
Algorithme ex01;
variable
    N, M, i, j, nb, S : entier;
    A: tableau de taille [20,30] : entier;
    B: tableau de taille [N]: entier;   T: tableau de taille [N]; entier;
    j                                                          [N]; entier;
debut
  1) lire(N, val) nb:=0;
   pour i:=1 à N
   faire lire[B(i)];
    si B[i]>0   alors   x:=B[i]
                        sinon   x:=-B[i];
    fsi;
   tq x>=val
   faire
    x:=x-val;
   fait;
  si x<>0 alors nb:=nb+1
  fait;
  écrire(nb);
  2) pour i:=1 à 20
     faire S:=0
       pour j:=1 à 30
```



faire lire A[i,j];
si A[i,j] < nb alors S := S + A[i,j];
fait;
K[i] := S;
fait;
Pour i := 1 à 20
faire écrire (K[i]);
fait;
3) pour j := 1 à 30
faire lire A[i,j], y := A[1,j];
pour i := 1 à 20
faire
si A[i,j] < y alors y := A[i,j]
fait;
T[j] := y
fait;
pour j := 1 à 30
faire
écrire (T[j]);
fait;
4)